DEVELOPMENT AND EVALUATION OF AN ORDER-N FORMULATION FOR MULTI-FLEXIBLE BODY SPACE SYSTEMS

Tushar K. Ghosh
L-3 Communications, STRATIS
1002, Gemini Avenue, Suite 200
Houston, TX 77058, USA
E-mail: tushar.ghosh@l-3com.com

Leslie J. Quiocho
Mail Code ER7
NASA Lyndon B. Johnson Space Center
2101 NASA Parkway
Houston, TX, 77058, USA
E-mail: leslie.j.quiocho@nasa.gov

KEYWORDS

Multibody dynamics, Rigid and flexible bodies, Tree topology, Constraints, Closed loops, Order-N.

ABSTRACT

This paper presents development of a generic recursive Order-N algorithm for systems with rigid and flexible bodies, in tree or closed-loop topology, with N being the number of bodies of the system. Simulation results are presented for several test cases to verify and evaluate the performance of the code compared to an existing efficient dense mass matrix-based code. The comparison brought out situations where Order-N or mass matrix-based algorithms could be useful.

INTRODUCTION

The Software, Robotics and Simulation Division (SRSD) of NASA Lyndon B. Johnson Space Center provides math modeling and simulation in support of engineering analyses and crew training activities for the center. The division currently has an efficient generic multibody dynamics code based on a dense mass-matrix formulation, which is used for simulating systems involving on-orbit robotic manipulators such as the Canadian Space Agency-built Space Station Manipulator System (SSRMS). It is generally known that Order-N (O(N)) algorithms, which involve arithmetic operation counts of the order N, where N is the number of bodies, perform more efficiently for systems with large degrees of freedom, compared to mass matrix-based with operations of order N^3. It was therefore decided to develop an O(N) simulation for SRSD to investigate applications where they may perform better. This development was performed in-house to allow maximum flexibility and control in different simulations.

ALGORITHM DEVELOPMENT

There are several methods in the literature that may be used for developing an O(N) algorithm. References may be found in (Banerjee, 2003). The formulation presented here is based on algebraically putting together the following steps: (1) kinematic equations relating motions between consecutive joints, (2) equations of motion of a single body, rigid or flexible, (3) equations relating the total spatial force and active forces and moments at the joint and (4) constraint conditions.

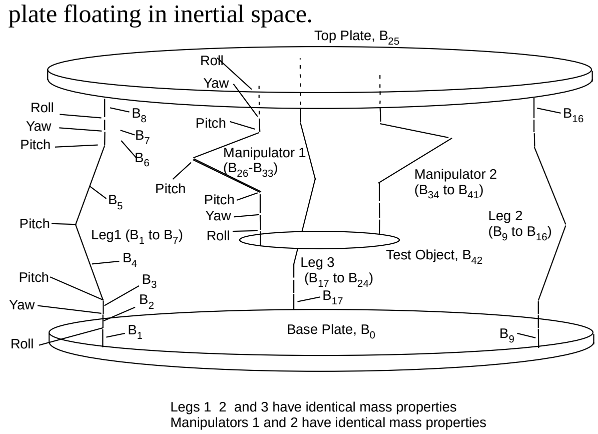
Derivation of the Order-N algorithm is presented in the Appendix. The effect of motion-induced stiffness (Banerjee, 1993) in flexible bodies of the system has not been incorporated yet. Inter-body forces may cause this effect to be important in some cases even for slow motions typical of space systems.

CODE VERIFICATION

The Order-N code was verified against the existing mass matrix-based code for several test cases, which itself was verified against other simulations in the industry, including TREETOPS (Singh et. al.,1985).The mass-matrix algorithm has been in use for many facilities at the Johnson Space Center for many years. Results from the two implementations matched with high accuracy (i.e., within 1.0e-10 or better).

SIMULATION TEST CASES

The test cases are based on variations of the system shown in Figure 1. The base plate B_0 is a rigid circular plate floating in inertial space.



Figures 1: Test Model Description

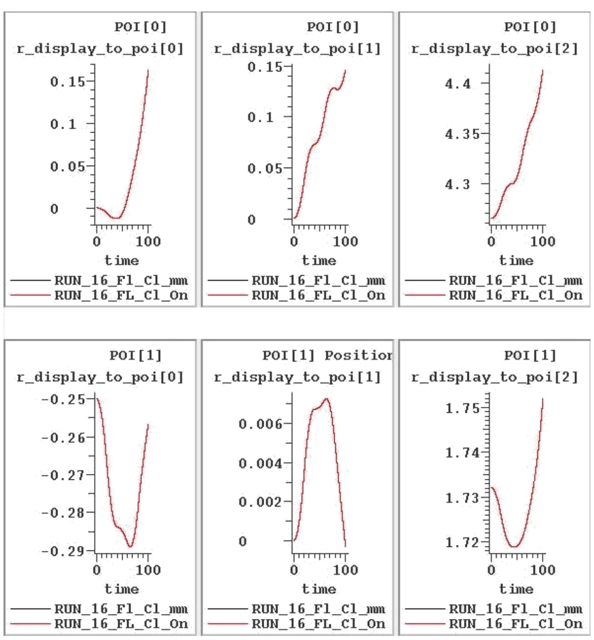
Three identical articulating legs and two identical manipulator arms are rigidly attached to the top plate. The other ends of the legs are rigidly connected to the base plate and the other ends of the manipulators hold a test object rigidly. All links are modeled as cylindrical rods. All joints of the legs and manipulators are single axis rotational joints. Two configurations of the legs and manipulators are considered, one with six joints the other with seven. The axes of the seven jointed manipulators and legs are in the order roll, yaw, pitch, pitch, pitch, yaw

and roll. The axes for ones with six joints are in the order roll, yaw, pitch, pitch, yaw and roll. In several configurations the boom elements of the legs have been split into two parts of equal length and joined rigidly, for adding additional bodies and flex degrees of freedom to the system. Only the boom elements are modeled as flexible. Each flexible rod has four bending modes (two in each of the bending planes). The system was driven by forces and moments on the top plate and the test object, and moments on joints of the first leg. The forces and moments were held constant for every 20 seconds and then switched in sign. Table 1 shows the configurations. Open loop cases are obtained by freeing the joints between legs 2, 3 and top plate and between manipulator 2 and the test object. For closed loop simulations constraint forces were determined at these same points.

Table 1: System Test Case Configurations

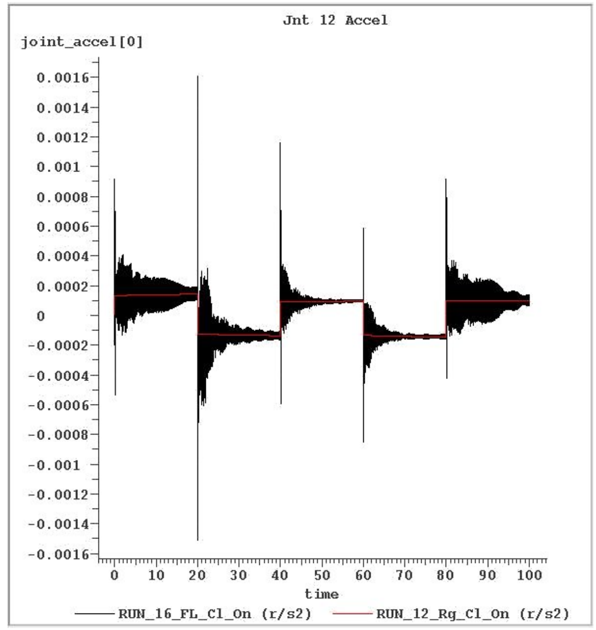
System Configuration	Legs			Manipulators		Total Number of Bodies
	Number of Bodies	Number of Joints	Boom Construction	Number of bodies	Number of Joints	
1	6	6	One rod	0	0	20
2	8	6	Two rods	0	0	26
3	10	7	Two rods	0	0	32
4	10	7	Two rods	8	7	49

RESULTS



Figures 2: Displacements of Top Plate and Test Object

Plots of representative data are shown in Figures 2 and 3. Figure 2 shows a co-plot of displacements of the center of the top plate, and the joint between manipulator 1 and the test object obtained from O(N) and mass matrix simulations for Case 16. The two results match to within 1.0e-10. Figure 3 is a co-plot of angular acceleration of the bottom pitch joint of leg 2 for Cases 12 and 16 (rigid and flex respectively) for O(N). Distances are shown in meters, while angles are shown in radians.



Figures 3: Leg 2 Bottom Pitch Joint Angular Acceleration

Performance of the O(N) code is measured by the CPU time for a 100 second simulation and comparing it with the existing code. The results of the comparison are listed below for rigid and flexible models separately for both open-loop and closed-loop scenarios. Rigid cases were run with 0.001 second and flexible cases were run with 0.0001 second integration time step. An Euler-Cromer integration scheme was used.

Table 2: Timing Results

Case Number	Rigid (R) or Flex (F) Simulation	Configuration (Table 1)	Open (O) or Closed(C) loop	Degrees of Freedom			CPU Time (sec)	
				Rigid	Flex	Total	Order-N	Mass Matrix
1	R	1	O	24	0	24	10.1	6.7
2	R	2	O	24	0	24	12.4	7.9
3	R	3	O	27	0	27	15.0	10.1
4	R	4	O	42	0	42	22.3	18.6

5	F	1	O	24	24	48	166.8	174.0
6	F	2	O	24	48	72	277.9	423.6
7	F	3	O	27	48	75	275.9	478.0
8	F	4	O	42	80	112	408.6	1127.9
9	R	1	C	24	0	24	40.7	17.1
10	R	2	C	24	0	24	48.0	18.4
11	R	3	C	27	0	27	59.4	22.7
12	R	4	C	42	0	42	133.8	57.4
13	F	1	C	24	24	48	613.4	567.6
14	F	2	C	24	48	72	937.2	1402.0
15	F	3	C	27	48	75	1037.9	1551.2
16	F	4	C	42	80	112	2257.3	4063.9

Cases 9, 10, 11, 13, 14 and 15 have two loops, cases 12 and 16 have three loops.

DISCUSSION AND CONCLUSIONS

The timing results confirm that the dense mass matrix formulation is faster than the O(N) formulation for smaller degrees of freedom (DOF's) and O(N) is faster otherwise. In our cases for all rigid systems mass-matrix take less CPU time than O(N) for both open-loop and closed-loop systems because they have fewer DOFs. Flexible body systems have more DOF and O(N) run faster, except for the case 13 involving two loops and fewer DOF than the other flex cases. Loops added large amount of computation and more so for O(N). For systems with loops the advantage of O(N) is reduced. With more loops the reduction is more because it requires solution of coupled linear constraint equations which can have a large number of unknown constraint forces and moments, and are solved in the usual manner that requires n_c^3 arithmetic operations, where n_c is the number of constrained degrees of freedom.

The high degree of match between the mass-matrix and O(N) results is expected because the two methods solve the same set of equations using many common codes.

ACKNOWLEDGMENT

The authors acknowledge the support provided by An Huynh and Thomas Brain of METECS Corporation, Houston Texas, USA in coding the simulation, and to Arun Banerjee, formerly of Lockheed Martin Palo Alto Research Laboratory, California, USA for review of the manuscript.

REFERENCES

Banerjee, A. K, "Block-Diagonal Equations for Multibody Elastodynamics with Geometric Stiffness and Constraints", *Journal of Guidance, Control and Dynamics*, Vol. 16, No. 6, 1993.

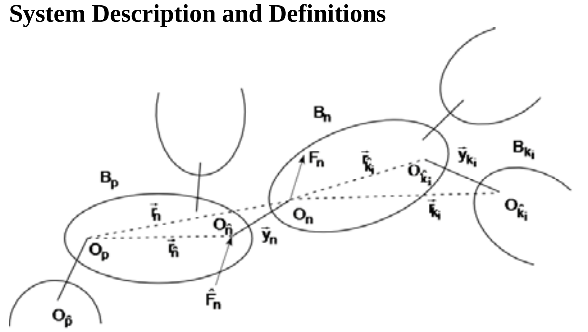
Banerjee, A. K., "Contributions of Multibody Dynamics to Space Flight: A Brief Review", *Journal of Guidance, Control and Dynamics*, Vol. 26, No. 3, 2003.

Quicho, L,J.; 2010, T. Ghosh, D. frenkel, A. Huynh, "Mode Selection Techniques in Variable Mass Flexible Modeling", In *AIAA Meeting Papers AIAA Modeling and Simulation Technologies Conference, Aug 2010, Toronto, Canada, AIAA-2010-7606.*

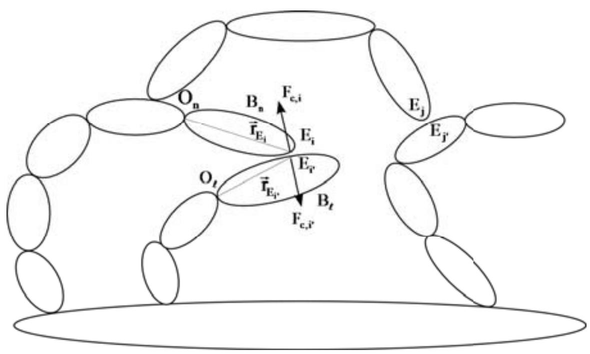
Singh, R.P., van der Voort, R.J., and Likins, P.W., "Dynamics of Flexible Bodies in Tree Topology-A Computer Oriented Approach", *Journal of Guidance, Control and Dynamics*, Vol.26, No. 3, 1985.

APPENDIX: DERIVATION OF O(N) EQUATIONS

System Description and Definitions



Figures A1: Multibody System Definitions



Figures A2: Constrained Motion Definitions

Consider the set of rigid and/or flexible bodies, in Figure A1, with the associated labeling of forces and points. $\mathcal{R}_n$ and $\mathcal{R}_{\hat{k}_i}$ are frames attached to body n (B_n) at its inboard and outboard joints O_n and $O_{\hat{k}_i}$ respectively. Figure A2 shows the labeling for constrained systems. Throughout this derivation it is assumed that all the vectors and inertia matrices are either generated in or are converted to a single reference frame, which is the structural reference frame $\mathcal{R}_0$ of the base body of the multibody system. The conversion is performed every time the system states are updated.

Kinematics of Motion between O_p and O_n

Let $\vec{v}_n, \vec{a}_n$ represent the inertial velocity and acceleration respectively, of O_n, $\vec{\omega}_n$ represent the inertial angular

velocity of $\mathcal{R}_n$ and η_p represent the flexible variables of the body B_p. Kinematic equations relating position, velocity and acceleration of O_p and $O_{\hat{n}}$, and angular velocity and angular acceleration of frames $\mathcal{R}_p$ and $\mathcal{R}_{\hat{n}}$ are given by

$$\vec{r}_{\hat{n}} = \vec{\rho}_{\hat{n}} + \phi_{\hat{n}}\eta_p,\; v_{\hat{n}} = \vec{v}_p + \tilde{\omega}_p r_n + \phi_{\hat{n}}\eta_p,$$

$$\vec{\omega}_{\hat{n}} = \vec{\omega}_p + \psi_{\hat{n}}\dot{\eta}_p$$

$$\dot{\vec{\omega}}_{\hat{n}} = \dot{\vec{\omega}}_p + \psi_{\hat{n}}\ddot{\eta}_n + \tilde{\omega}_p\psi_{\hat{n}}\dot{\eta}_n$$

$$\vec{a}_{\hat{n}} = \vec{a}_p - \tilde{r}_n\dot{\vec{\omega}}_p + \phi_{\hat{n}}\ddot{\eta}_p + \tilde{\omega}_p\tilde{\omega}_p\vec{r}_n + 2\tilde{\omega}_p\phi_{\hat{n}}\dot{\eta}_p$$

$\vec{\rho}_p$ is the undeformed value of $\vec{r}_p$, the vector from O_p to $O_{\hat{n}}$. Combining the last two equations we can write

$$A_{\hat{n}} = \mathcal{F}_{\hat{n},p}A_p + S_{\hat{n}}\ddot{\eta}_n + A_{\hat{n},r} \tag{1.1}$$

where, $A_{\hat{n}} = \begin{Bmatrix} \vec{a}_{\hat{n}} \\ \dot{\vec{\omega}}_{\hat{n}} \end{Bmatrix}$, $\mathcal{F}_{n,\hat{n}} = \begin{bmatrix} 1 & -\tilde{r}_{\hat{n}} \\ 0 & 1 \end{bmatrix}$, $S_n = \begin{bmatrix} \phi_{\hat{n}} \\ \psi_{\hat{n}} \end{bmatrix}$

and $A_{\hat{n},r} = \begin{Bmatrix} \tilde{\omega}_p\tilde{\omega}_p\vec{r}_{\hat{n}} + 2\tilde{\omega}_p\phi_{\hat{n}}\dot{\eta}_p \\ \tilde{\omega}_p\,\psi_{O,n}\dot{\eta}_n \end{Bmatrix}$.

$\mathcal{F}$ denotes a shift operator corresponding to an offset between two points identified by the subscripts. In this case the offset is $\vec{r}_{\hat{n}}$. The tilde (~) is the usual cross product operator on a vector.

The joint n may have up to six DOFs allowing both translation and rotation. The translation of O_n with respect to $O_{\hat{n}}$ is given by $\vec{y}_n = \sum_{i=1}^{NT_n} \hat{g}_{i,n}\delta_{i,n} = G_n\delta_n$ where $\hat{g}_{i,n}$ are linearly independent unit vectors fixed in $O_{\hat{n}}$, $NT_n \le 3$ is the number of translational degrees of freedom of joint n and $\delta_{i,n}$ are scalar quantities, representing joint translations in the directions of the unit vectors. The inertial acceleration of O_n is:

$$\vec{a}_n = \vec{a}_{\hat{n}} - \tilde{y}_n\dot{\vec{\omega}}_{\hat{n}} + G_n\ddot{\delta}_n + \vec{a}_n^{ry} \tag{1.2}$$

where, $\vec{a}_n^{ry} = \tilde{\omega}_{\hat{n}}\tilde{\omega}_{\hat{n}}\vec{y}_n + 2\,\tilde{\omega}_{\hat{n}}G_n\dot{\delta}_n$

Inertial angular velocity of $\mathcal{R}_n$ is given by

$$\vec{\omega}_n = \vec{\omega}_{\hat{n}} + \sum_{i=1}^{NR_n} \hat{l}_{i,n}\dot{\theta}_{i,n} = \vec{\omega}_{\hat{n}} + L_n\dot{\theta}_n$$

where $\hat{l}_{i,n}$ is the unit vector in the direction of the rotation $\theta_{i,n}$ and NR_n is the number of rotational degrees of freedom of the joint. L_n is a matrix whose columns are the unit vectors $\hat{l}_{i,n}$ and θ_n is a column matrix containing the rotations $\theta_{i,n}$. Differentiating $\vec{\omega}_n$ in the inertial frame we get

$$\dot{\vec{\omega}}_n = \dot{\vec{\omega}}_{\hat{n}} + L_n\ddot{\theta}_n + \vec{\alpha}_n^{r\theta} \tag{1.3}$$

$$\vec{\alpha}_n^{r\theta} = \dot{L}_n\dot{\theta}_n$$

Combining Equations (1.2) and (1.3),

$$A_n = \mathcal{F}_{n,\hat{n}}A_{\hat{n}} + P_n\ddot{\gamma}_n + A_n^{rj} \tag{1.4}$$

$$A_n = \begin{Bmatrix} \vec{a}_n \\ \dot{\vec{\omega}}_n \end{Bmatrix},\; \ddot{\gamma}_n = \begin{Bmatrix} \ddot{\delta}_n \\ \ddot{\theta}_n \end{Bmatrix},\; A_n^{rj} = \begin{Bmatrix} \vec{a}_n^{ry} \\ \vec{\alpha}_n^{r\theta} \end{Bmatrix}$$

$$\mathcal{F}_{n,\hat{n}} = \begin{bmatrix} 1 & -\tilde{y}_n \\ 0 & 1 \end{bmatrix} \text{ and } P_n = \begin{bmatrix} G_n & 0 \\ 0 & L_n \end{bmatrix} \tag{1.5}$$

Finally, combining Equations (1.1) and (1.4) we get

$$A_n = \mathcal{F}_{n,p}A_p + S_n\ddot{\eta}_p + P_n\ddot{\gamma}_n + A_n^r \tag{1.6}$$

where, using the relationship $\vec{r}_n = \vec{r}_{\hat{n}} + \vec{y}_n$,

$$\mathcal{F}_{n,p} = \mathcal{F}_{n,\hat{n}}\mathcal{F}_{\hat{n},p},\; S_n = \mathcal{F}_{n,\hat{n}}S_{\hat{n}},\; A_n^r = \mathcal{F}_{n,\hat{n}}A_{\hat{n}}^r + A_n^{rj}$$

Inter-Body and Joint Actuator Forces

Let F_n be the spatial forces exerted by B_p on B_n at O_n and $\hat{F}_{\hat{n}}$ be the spatial force exerted by B_n on B_p at $O_{\hat{n}}$. Then from equilibrium considerations,

$$\hat{F}_{\hat{n}} = -\mathcal{F}_{n,\hat{n}}^T F_n \tag{2.1}$$

$\mathcal{F}_{n,\hat{n}}$ is given by Equation (1.5). Let $\mu_{i,n}$ and $\nu_{j,n}$ be respectively, the i-th actuator force $(i \le 3)$ and j-th actuator torque $(j \le 3)$ on body n at O_n. They act in the directions of the degrees of freedom $\hat{g}_{i,n}$ and $\hat{l}_{j,n}$ of the joint. Defining $\sigma_n = \begin{bmatrix} \mu_n \\ \nu_n \end{bmatrix}$ as the array of actuator forces and moments at joint n, it is straight forward to show that

$$\sigma_n = P_n^T F_n \tag{2.2}$$

where P_n is given by Equation (1.5).

Equations of Motion of Body n

A minor modification of equations of a flexible body in Quiocho, et. al (Quiocho, 2010), produces the equations of a body B_n in rigid and flexible coordinates as

$$M_{rr,n}A_n + M_{re,n}\ddot{q}_n = \sum_i \mathcal{F}_{i,n}^T F_{e,i,n} + B_{r,n}$$

$$M_{er,n}A_n + M_{ee,n}\ddot{q}_n + K_{ee,n}q_n + D_{ee,n}\dot{q}_n = \sum_i S_{i,n}^T F_{e,i,n} + B_{e,n}$$

We shall write the equations of motion of the body n after separating the external spatial forces $F_{e,i,n}$ on B_n as (i) from the previous body acting at the inboard joint (F_n), (ii) from child bodies B_{k_i} acting at outboard joints ($\hat{F}_{\hat{k}_i}$), (iii) forces at constrained points ($F_{c,j}$) when there are such points on the body, and (iv) external forces ($F_{ext,i,n}$) acting on the body. Because we chose the reference frame of the body to be at the inboard joint, the shift function at the inboard joint is an identity matrix and the shape/slope function at the inboard joint is a null matrix. Equations for body n are:

$$M_{rr,n}A_n + M_{re,n}\ddot{q}_n = F_n + \sum_i \mathfrak{F}^T_{\hat{k}_i,n}\hat{F}_{\hat{k}_i} + \sum_{j\in Z_c(n)} \mathfrak{F}^T_{c,j}F_{c,j} + \sum_i \mathfrak{F}^T_{i,n}F_{ext,i,n} + B_{r,n}$$

$$M_{er,n}A_n + M_{ee,n}\ddot{q}_n + K_{ee,n}q_n + D_{ee,n}\dot{q}_n = \sum_i S^T_{\hat{k}_i}\hat{F}_{\hat{k}_i} + \sum_{j\in Z_c(n)} S^T_{c,j}F_{c,j} + \sum_i S^T_{i,n}F_{ext,i,n} + B_{e,n}$$

In the above equations $F_{c,j}$ is the spatial force at the j-th constrained point, $\mathfrak{F}_{c,j}$ is the shift function for the offset of the constrained point from O_n and $S_{c,j}$ is the shape/slope function of body n at the constrained point.

Defining

$$G_{r,n} = \sum_i \mathfrak{F}^T_{i,n}F_{ext,i,n} + B_{r,n} \tag{3.1}$$

$$G_{e,n} = \sum_i S^T_{i,n}F_{ext,i,n} + B_{e,n} - K_{ee,n}q_n - D_{ee,n}\dot{q}_n \tag{3.2}$$

Using Equations (2.1) for $\hat{F}_{\hat{k}_i}$ and then Equations (3.1) and (3.2) we get the equations of motion of the body n as

$$M_{rr,n}A_n + M_{re,n}\ddot{q}_n = F_n - \sum_i \mathfrak{F}^T_{k_i,n}F_{k_i} + \sum_{j\in Z_c(n)} \mathfrak{F}^T_{c,j}F_{c,j} + G_{r,n} \tag{3.3}$$

$$M_{er,n}A_n + M_{ee,n}\ddot{q}_n = -\sum_i S^T_{k_i}F_{k_i} + \sum_{j\in Z_c(n)} S^T_{c,j}F_{c,j} + G_{e,n} \tag{3.4}$$

Here, $Z_c(n)$ is the set of indices for constrained points on body n.

Recursive Solution of Equations of Motion

The dynamical equations of motion of the system are solved recursively in steps as follows.

<u>Step 1</u>. Forward Pass Kinematics: In this step all position and velocity states are generated starting with the base body using equations derived in Kinematics section.

Step 2. Inward Pass Dynamics Equations:

Observing the equations of motion it is reasonable to expect that for any n F_n can be expressed as linear functions of A_p, $\ddot{q}_p$ and constraint forces on itself and on bodies on outer branches:

$$F_n = D_{A,n}A_p + D_{q,n}\ddot{q}_p + d_n + \sum_{j\in Y_c(n)} D_{c,n,j}F_{c,j} \tag{4.1}$$

$Y_c(n)$ is the set of indices of all constrained points on body n and on all bodies of its outer branches.

We shall determine the coefficients in the above equations recursively. In Euation (4.1) using k_i in place of n and n in place of p, and using the result in Equation (4.1) we get

$$\ddot{q}_n = \hat{Q}_{A,n}A_n + \hat{q}_n + \sum_{j\in Y_c(n)} \hat{Q}_{c,n,j}F_{c,j} \tag{4.2}$$

$$\hat{Q}_{A,n} = \hat{M}^{-1}_{ee,n}\hat{M}_{er,n}, \quad \hat{M}_{ee,n} = M_{ee,n} + \sum_{i\in Z(n)} S^T_{k_i}D_{q,k_i}$$

$$\hat{M}_{er,n} = M_{er,n} + \sum_{i\in Z(n)} S^T_{k_i}D_{A,k_i}$$

$$\hat{q}_n = M^{-1}_{ee,n}[G_{e,n} - \sum_{i\in Z(n)} S^T_{k_i}d_{k_i}]$$

$$\hat{Q}_{c,n,j} = -\hat{M}_{ee,n}S^T_{k_i}D_{c,k_i,j} \quad \text{for } i\in Z(n),\ j\in Y_c(k_i)$$

$$\hat{Q}_{c,n,j} = -\hat{M}_{ee,n}S^T_{c,j} \quad \text{for } j\in Z_c(n)$$

$Z(n)$ is the set of indices of child bodies of B_n. In the same manner, substituting for F_{k_i} in Eq. (3.3) and rearranging we get,

$$\overline{M}_{rr,n}A_n + \overline{M}_{re,n}\ddot{q}_n = F_n + \overline{G}_{r,n} + \sum_{j\in Y_c(n)} \overline{D}_{c,n,j}F_{c,j} \tag{4.3}$$

$$\overline{M}_{rr,n} = M_{rr,n} + \sum_{i\in Z(n)} \mathfrak{F}^T_{k_i,n}D_{A,k_i},$$

$$\overline{M}_{re,n} = M_{re,n} + \sum_{i\in Z(n)} \mathfrak{F}^T_{k_i,n}D_{q,k_i},$$

$$\overline{G}_{r,n} = G_{r,n} - \sum_{i\in Z(n)} \mathfrak{F}^T_{k_i,n}d_{k_i}`$$

$$\overline{D}_{c,n,j} = -\mathfrak{F}^T_{k_i,n}D_{c,k_i,j} \text{ for } i\in Z(n),\ j\in Y_c(k_i)$$

$$\overline{D}_{c,n,j} = \mathfrak{F}^T_{c,j} \text{ for } j\in Z_c(n)$$

Using Equation (4.2) for $\ddot{q}_n$ in Equation (4.3) we get

$$\hat{M}_{rr,n}A_n = F_n + \hat{G}_{r,n} + \sum_{j\in Y_c(n)} \hat{D}_{c,n,j}F_{c,j} \tag{4.4}$$

$$\hat{M}_{rr,n} = \overline{M}_{rr,n} + \overline{M}_{re,n}\hat{Q}_{A,n}, \quad \hat{G}_{r,n} = \overline{G}_{r,n} - \overline{M}_{re,n}\hat{q}_n$$

$$\hat{D}_{c,n,j} = \overline{D}_{c,n,j} - \overline{M}_{re,n}\hat{Q}_{c,k_i,j}$$

Using Equation (1.6) for A_n in Equation (4.4) and rearranging, we get

$$\hat{M}_{rr,n}\mathfrak{F}_{n,p}A_p + \hat{M}_{rr,n}S_{n,p}\ddot{q}_p + \hat{M}_{rr,n}P_n\ddot{\gamma}_n = F_n + \hat{G}_{r,n} - \hat{M}_{rr,n}A_{n,R} + \sum_{j\in Y_c(n)} \hat{D}_{c,n,j}F_{c,j}$$

Pre-multiplying this equation by P^T_n, using $\sigma_n = P^T_nF_n$ from Equation (2.2) and solving the resulting equation for $\ddot{\gamma}_n$ we get

$$\ddot{\gamma}_n = B_{A,n}A_p + B_{q,n}\ddot{q}_p + b_n + \sum_{j\in Y_c(n)} B_{c,n,j}F_{c,j} \tag{4.5}$$

$$B_{A,n} = -M^{-1}_{\gamma\gamma,n}P^T_n\hat{M}_{rr,n}\mathfrak{F}_{n,p},$$

$$B_{q,n} = -M^{-1}_{\gamma\gamma,n}P^T_n\hat{M}_{rr,n}S_{n,p},$$

$$b_n = -M^{-1}_{\gamma\gamma,n}\left[\sigma_n + P^T_n(\hat{G}_{r,n} - \hat{M}_{rr,n}A_{n,R})\right],$$

$$B_{c,n,j} = -M^{-1}_{\gamma\gamma,n}P^T_n\hat{D}_{c,n,j}, \text{ and } M_{\gamma\gamma,n} = P^T_n\hat{M}_{rr,n}P_n$$

Equation (4.5) for $\ddot{\gamma}_n$ in Equation (1.6) yields for A_n

$$A_n = W_{A,n}A_p + W_{q,n}\ddot{q}_p + w_n + \sum_{j\in Y_c(n)} W_{c,n,j}F_{c,j} \tag{4.6}$$

$W_{A,n} = \mathfrak{F}_{n,p} + P_n B_{A,n}$, $W_{q,n} = S_{n,p} + P_n B_{q,n}$,
$w_n = P_n b_n + A_{n,R}$, and $W_{c,n,j} = P_n B_{c,n,j}$

Equation (4.6) for A_n in Equation (4.2) gives

$$\ddot{q}_n = Q_{A,n} A_p + Q_{q,n} \ddot{q}_p + q_n + \sum_{j \in Y_c(n)} Q_{c,n,j} F_{c,j} \tag{4.7}$$

$Q_{A,n} = \hat{Q}_{A,n} W_{A,n}$, $Q_{q,n} = \hat{Q}_n W_{q,n}$, $\hat{q}_n = \hat{Q}_{A,n} w_n + \hat{q}_n$, and $Q_{c,n,j} = \hat{Q}_{A,n} W_{c,n,j} + \hat{Q}_{c,n,j}$

Finally, using Equation (4.6) in Equation (4.4) we get

$$F_n = D_{A,n} A_p + D_{q,n} \ddot{q}_p + d_n + \sum_{j \in Y_c(n)} D_{c,n,j} F_{c,j} \tag{4.8}$$

$D_{A,n} = \hat{M}_{rr,n} W_{A,n}$, $D_{q,n} = \hat{M}_{rr,n} W_{q,n}$
$d_n = \hat{M}_{rr,n} w_n - \hat{G}_{r,n}$, and
$D_{c,n,j} = \hat{M}_{rr,n} W_{c,n,j} - \hat{D}_{c,n,j}$

We can see that Equation (4.8) is in the same form as Equation (4.1) we started with, confirming that if the latter is true for child bodies, it would be true for the current body also. Following the same steps as above for any outermost body for which $F_{\hat{k}_i} = 0$ it is easy to show that Equation (5.1) is true for such bodies. Using the induction logic it therefore follows that Equation (4.1) is true for all bodies of the system.

Computations for systems with multiple branches need to start from an outermost body, move inward till a body with multiple child bodies is reached. Inward computation from a body with multiple child bodies should continue only after computations for all of its child bodies are completed.

A_0 and F_0 for the Base Body B_0 in terms of the constraint forces are obtained from Equation (4.4). When B_0 is fixed in inertial frame $A_0 = 0$ and we then have

$$F_0 = -\hat{G}_{r,0} + \sum_{j \in Y_c(0)} \hat{D}_{c,0.j} F_{c,j}$$

When B_0 is free in the inertial frame $F_0 = 0$ and

$$A_0 = \hat{M}_{rr,0}^{-1} [\hat{G}_{r,0} + \sum_{j \in Y_c(0)} \hat{D}_{c,0.j} F_{c,j}] \tag{4.9}$$

The flexible coordinate acceleration $\ddot{q}_0$ for the base body is obtained from Equation (4.2)

$$\ddot{q}_0 = \hat{Q}_{A,0} A_0 + \hat{q}_0 + \sum_{j \in Y_c(n)} \hat{Q}_{c,n,j} F_{c,j} \tag{4.10}$$

For systems without closed loops the $F_{c,j}$ term drops out and the equations obtained above are sufficient for determining A_0 and $\ddot{q}_0$, and then the system accelerations, by successive computation of $\ddot{\gamma}_n$, $\ddot{q}_n$ and A_n in an outward sweep.

Step 3. Accelerations of Constrained Points in terms of Forces at Constrained Points:

For the determination of acceleration of constrained points in terms of forces at these points we seek to express the accelerations A_n and $\ddot{q}_n$ for bodies in the path from base body to the constrained points in the form

$$A_n = h_n^A + \sum H_{n,j}^A F_{c,j} \quad \text{and} \quad \ddot{q}_n = h_n^q + \sum H_{n,j}^q F_{c,j} \tag{4.11}$$

Here the summation index j covers all constrained points. It follows from Equations (4.9) and (4.11) that when the base is free in inertial frame

$h_0^A = \hat{M}_{rr,0}^{-1} \hat{G}_{r,0}$, $H_{0,j}^A = \hat{M}_{rr,0}^{-1} \hat{D}_{c,0,j}$, $h_0^q = \hat{Q}_{A,0} h_0^A + \hat{q}_0$ and $H_0^q = \hat{Q}_{c,0,j}$.

When the base is fixed, $h_0^A = 0$, $H_{0,j}^A = 0$, $h_0^q = \hat{q}_0$ and $H_0^q = \hat{Q}_{c,0,j}$.

Using Equations (4.7) and (4.11) we get the recursive equations for h_n^q and $H_{n,j}^q$:

$h_n^q = Q_{A,n} h_p^A + Q_{q,n} h_p^q + q_n$ and
$H_{n,j}^q = Q_{A,n} H_{p,j}^A + Q_{q,n} H_{p,j}^q + Q_{c,n,j}$

Using Equation (4.6) and (4.11) we get the recursive equations for h_n^A and $H_{n,j}^A$:

$h_n^A = W_{A,n} h_p^A + W_{q,n} h_p^q + w_n$, and
$H_{n,j}^q = W_{A,n} H_{p,j}^A + W_{q,n} H_{p,j}^q + W_{c,n,j}$.

Let E_i be the point in constraint i located on body n and $E_{i'}$ be the mating point of the same constraint, located on body ℓ . The spatial acceleration of E_i is

$$A_{E_i} = \mathfrak{F}_{E_i,n} A_n + S_{E_i,n} \ddot{q}_n + A_{E_i,r} \tag{4.12}$$

where $\mathfrak{F}_{E_i,n}$ is the standard shift operator for the offset $\vec{r}_{E_i}$ of E_i with respect to O_n $S_{E_i,n} = \begin{bmatrix} \phi_{E_i} \\ \psi_{E_i} \end{bmatrix}$ is the shape/slope function of B_n at E_i and,

$$A_{E_i,r} = \begin{bmatrix} \tilde{\omega}_n (\tilde{\omega}_n \vec{r}_{E_i} + 2\phi_{E_i} \dot{q}_n) \\ \tilde{\omega}_n \psi_{E_i} \dot{q}_n \end{bmatrix}.$$

For each constraint k (k = 1, 2, …, n_c) the constraint forces $F_{c,k}$ and $F_{c,k'}$ at the two constrained points E_k and $E_{k'}$ respectively are equal and opposite, i.e, $F_{c,k'} = -F_{c,k}$. Using this and Equation (4.11) for A_n and $\ddot{q}_n$ in Equation (4.12) we get,

$$A_{E_i} = \mathfrak{F}_{E_i,n} h_n^A + S_{E_i,n} h_n^q + A_{E_i,r} + \sum_{k=1}^{n_c} [\mathfrak{F}_{E_i,n} (H_{n,k}^A - H_{n,k'}^A) + S_{E_i,n} (H_{n,k}^q - H_{n,k'}^q)] F_{c,k} \tag{4.13}$$

Similarly, spatial acceleration of $E_{i'}$ is given by

$$A_{E_{i'}} = \mathfrak{F}_{E_{i'},n} h_\ell^A + S_{E_{i'},\ell} h_\ell^q + A_{E_{i'},r} + \sum_{k=1}^{n_c} [\mathfrak{F}_{E_{i'},\ell} (H_{\ell,k}^A - H_{\ell,k'}^A) + S_{E_{i'},\ell} (H_{\ell,k}^q - H_{\ell,k'}^q)] F_{c,k}$$

(4.14)

Step 4. Determination of Constraint Forces**:**

Using equations (4.13) and (4.14) the difference in the accelerations at the constraint i may be written as

$$\Delta A_{E_i} = A_{E_i} - A_{E_{i'}} = \Delta h_{E_i} + \sum_k \Delta H_{E_i,k} F_{c,k} \qquad (4.15)$$

$$\Delta h_{E_i} = \mathfrak{F}_{E_{i'},\ell} h_\ell^A + S_{E_{i'},\ell} h_\ell^q + A_{E_{i'},r} - \mathfrak{F}_{E_i,n} h_n^A - S_{E_i,n} h_n^q - A_{E_i,r}$$

$$\Delta H_{E_i,k} = \mathfrak{F}_{E_{i'},\ell}(H_{\ell,k}^A - H_{\ell,k'}^A) - \mathfrak{F}_{E_i,n}(H_{n,k}^A - H_{n,k'}^A) + S_{E_{i'},\ell}(H_{\ell,k}^q - H_{\ell,k'}^q) - S_{E_i,n}(H_{n,k}^q - H_{n,k'}^q)$$

The summation range for k is all the constraints.

The linear and angular constraints at E_i are

$$\hat{g}_{c,i,j} \bullet (\vec{v}_{E_{i'}} - \vec{v}_{E_i}) = 0 \text{ for } 0 < j \le n_{t,i}$$

$$\hat{\ell}_{c,i,j} \bullet (\vec{\omega}_{E_{i'}} - \vec{\omega}_{E_i}) = 0 \quad \text{for } 0 < j \le n_{r,i}$$

where $\vec{v}$ and $\vec{\omega}$ represent the inertial velocity and angular velocity of points corresponding to the subscripts, $\hat{g}_{c,i,j}$ and $\hat{\ell}_{c,i,j}$ are unit vectors in the direction of translational and rotational constraints respectively, and $n_{t,i} (\le 3)$ and $n_{r,i} (\le 3)$ are the number of these constraints, respectively. $\vec{v}_{E_i}$, $\vec{v}_{E_{i'}}$, $\vec{\omega}_{E_i}$ and $\vec{\omega}_{E_i}$ are determined in the manner used for the point $O_{\hat{n}}$ in the Kinematics section,. Defining

$$\Delta V_{E_i} = \begin{bmatrix} \vec{v}_{E_{i'}} - \vec{v}_{E_i} \\ \vec{\omega}_{E_{i'}} - \vec{\omega}_{E_i} \end{bmatrix} \quad \text{and} \quad P_{c,i} = \begin{bmatrix} G_{c,i} & 0 \\ 0 & L_{c,i} \end{bmatrix} \quad \text{where}$$

$G_{c,i}$ is a $3 \times n_{t,i}$ matrix whose columns are the unit vectors $\hat{g}_{c,i,j}$ and $L_{c,i}$ is a $3 \times n_{r,i}$ matrix whose columns are the unit vectors $\hat{\ell}_{c,i,j}$, the difference in the spatial velocities of the constrained points at constraint i may be written as $P_{c,i}^T \Delta V_{E_i}$. The difference in the spatial acceleration in the constraint directions at the constraint point should be zero, giving

$$P_{c,i}^T \Delta A_{E_i} + \dot{P}_{c,i}^T \Delta V_{E_i} = 0$$

Using Baumgarte's stabilization scheme to limit constraint violation caused by numerical errors, this equation is modified to

$$P_{c,i}^T \Delta A_{E_i} + \dot{P}_{c,i}^T \Delta V_{E_i} + K_i P_{c,i}^T \Delta X_{E_i} + C_i P_{c,i}^T \Delta V_{E_i} = 0$$

K_i and C_i are positive constants to provide constraint stabilization. Using Equation (4.15) for ΔA_{E_i} we have

$$P_{c,i}^T \sum_k \Delta H_{E_i,k} F_{c,k} = -\dot{P}_{c,i}^T \Delta h_{E_i} - \dot{P}_{c,i}^T \Delta V_{E_i} - K_i P_{c,i}^T \Delta X_{E_i} - C_i P_{c,i}^T \Delta V_{E_i} \qquad (4.16)$$

Let us define $f_{c,k,j}$ to be the constraint force in the direction $\hat{g}_{c,k,j}$ and $\tau_{c,k,j}$ the constraint torque in the direction $\hat{\ell}_{c,k,j}$ and $\hat{F}_{c,k} = \begin{bmatrix} f_{c,k} \\ \tau_{c,k} \end{bmatrix}$. The force $F_{c,k}$ at the constraint point E_k due to forces and moments in the constraint directions is then given by $P_{c,k}\hat{F}_{c,k}$. Let $P_{f,k} = \begin{bmatrix} G_{f,k} & 0 \\ 0 & L_{f,k} \end{bmatrix}$ where $G_{f,k}$ and $L_{f,k}$ are made of the unit vectors normal to the constraint directions for translation and rotation and $\hat{F}_{f,k}$ be the array of forces and moments in these directions. Net spatial force at E_k is

$$F_{c,k} = P_{c,k}\hat{F}_{c,k} + P_{f,k}\hat{F}_{f,k} \qquad (4.17)$$

Restricting to cases where $\hat{F}_{f,k}$ is fully known, we use $F_{c,k} = P_{c,k}\hat{F}_{c,k} + P_{f,k}\hat{F}_{f,k}$ in Equation (4.17) to get

$$P_{c,i}^T \sum_k \Delta H_{E_i,k} P_{c,k} \hat{F}_{c,k} = Z_{c,i} \qquad (4.18)$$

$$Z_{c,i} = -P_{c,i}^T [\sum_k \Delta H_{E_i,k} P_{f,k} \hat{F}_{f,k} + \Delta h_{E_i}] - \dot{P}_{c,i}^T \Delta V_{E_i} - K_i P_{c,i}^T \Delta X_{E_i} - C_i P_{c,i}^T \Delta V_{E_i} \qquad (4.19)$$

Stacking Equation (4.18) for all constraints we get

$$M_c \hat{F}_c = Z_c \qquad (4.20)$$

where the $[i,k]$ submatrix of matrix M_c is given by

$$M_{c,i,k} = P_{c,i}^T \Delta H_{E_i,k} P_{c,k}$$

and Z_c is made of arrays $Z_{c,i}$ given by Equation (4.19). Equation (4.20) is solved for $\hat{F}_c$ and the spatial forces $F_{c,k}$ at constraint points are found using Equation (4.17).

Step 5. Computation of System Accelerations**:**

After determination of forces at the constrained points, A_0 and $\ddot{\eta}_0$ are determined using Equations (4.9) and (4.10). $\ddot{\gamma}_n$, $\ddot{q}_n$, and A_n are determined recursively in a forward pass using Equations (4.5), (4.7) and (1.6) respectively.